Prefiltering and Reconstruction Filters in the Volume Rendering

Alan Goluban*, Zeljka Mihajlovic**, Leo Budin**

***Abstract*—This paper deals with reconstruction filters and appropriate prefiltering techniques in the volume rendering. Sampling, resampling and reconstruction of continuous function are often used in many different fields. To satisfy conditions of Shannon theorem, prefiltering of images or data volumes is usually required before sampling. Usual approach is to use Gaussian prefilter, where the cutoff frequency is determined by the sampling rate, and one of different reconstruction schemes is then applied.**

The problem is to determine a prefilter in such a way that the reconstruction reproduce original object as accurately as possible. If the combination of prefiltering and reconstruction in the volumetric space is not adequate, aliasing artifacts will appear. Aliasing artifacts projects from the three-dimensional space onto the projection plane and superpose to the artifacts caused by sampling and reconstruction in two dimensions.

In this paper we present a methodology for designing filters based on optimization of combined operation of prefiltering and reconstruction postfiltering. This methodology offers a framework for comparing different reconstruction schemes in equal circumstances. The correct choice of reconstruction will also influence the derivative reconstruction which is used for shading. We demonstrate our results on examples in two- and three-dimensional space.

***Index Terms*-- interpolation, volume reconstruction, prefiltering, visualization.**

I. Introduction

In the volume rendering, the volumes are usually represented by a set of uniformly spaced sample data. One of the fundamental operations in visualization is reconstruction of a continuous function from a given set of samples. To reconstruct the value at arbitrary position different reconstruction approaches can be applied. Reconstruction approaches reach from the simplest such as nearest neighbor to trilinear interpolation, cubic splines e.g. BC-splines introduced by Mitchell and Netravali [1], Catmull-Rom spline [4], or B-spline approximation and interpolation [6]. The reconstruction scheme cause various appearances of aliasing artifacts. Aliasing introduced in three-dimensions will project in visualization process in the two-dimensions and superpose to the artifacts caused by sampling and reconstruction in two dimensions. Reconstruction of the derivative is also required. Derivative in the volume visualization is applied in shading to determine illumination. In illumination model derivative is used for the normal estimation, and although reconstruction can be correct, error in derivative reconstruction have high influence on the perception of the projected object.

Much work has been done towards the design and specification of reconstruction filters. However, before sampling, prefiltering is required and choice of the prefilter is not usually an argumented. Necessary condition for any kind of mathematical consideration is bandlimited condition for signal. Volume acquisition devices such as scanners or cameras, perform a lowpass filtering that bandlimits the function, but down-sampling is often required. Before down-sampling, lowpass prefiltering is needed to prevent aliasing caused by sampling. After sampling procedure, reconstruction of continuous function is required to perform resampling, up-sampling, reconstruction of derivative, or simply to reproduce the original object. In two-dimensions reconstruction can be done as convolution in x and y direction, and in the volume rendering reconstruction filter is incorporated in the resampling procedure along each ray.

Observation of impulse response of the reconstruction filter lead us to conclusion that application of high pass prefilter before reconstruction can improve final impulse response of the combined operation of the sampling and reconstruction. In this paper we propose methodology for designing lowpass prefilter that precede down-sampling, prefilter that is used for improvement of the final impulse response, and reconstruction filter. The methodology is based on optimization of combined operation of these filters.

II. Design of the Prefilter

For example, we will consider 2-dimensional case, starting with original set of samples $f(k,l)$. Next thing we want to do is to reduce number of samples for factor M in x direction, and factor N in y direction. Simple down-sampling is unacceptable, because this could generally introduce strong aliasing artifacts. So we want to find reduced set of new coefficients $y(k,l)$. To reconstruct the original samples $f(k,l)$, those coefficients must be up-sampled by factors M and N in x and y directions respectively. This step is followed by filtering with coefficients $b(k,l)$. Filtering sequence (reconstruction kernel) $b(k,l)$ is given in advance; for example discrete B-splines, Catmull-Rom spline, Gaussian function, e.g. any

*Department of Control and Computer Engineering in Automation,
**Department of Electronics, Microelectronics, Computer and Intelligent Systems,
University of Zagreb, Faculty of Electrical Engineering and Computing, Unska 3, 10000 Zagreb, Croatia
{alan.goluban, zeljka.mihajlovic, leo.budin}@fer.hr

sequence with "good" properties. Now, the goal is to calculate $y(k,l)$, so that reconstructed sequence is as close as possible to $f(k,l)$. For this purpose, we will consider the square difference between original and reconstructed sequence ε:

$$\varepsilon^2 = \sum_k \sum_l \left(f(k,l) - [y(k,l)]_{\uparrow M \uparrow N} * b(k,l) \right)^2 ,$$

where $\uparrow M \uparrow N$ denotes up-sampling by an integer multiple defined as:

$$[y(i,j)]_{\uparrow M \uparrow N} = \begin{cases} y\left(\frac{i}{M}, \frac{j}{N}\right), & i \text{ and } j \text{ multiple of } M \text{ and } N \\ 0 & \text{otherwise} \end{cases}$$

Generally, ε is a function of $y(p,q)$, for all possible values of p and q. At the point of minimum, all its partial derivatives with respect to $y(p,q)$ must vanish:

$$\begin{aligned} \frac{\partial \varepsilon^2}{\partial y(p,q)} &= -2 \sum_k \sum_l \left(\left(f(k,l) - [y(k,l)_{\uparrow M \uparrow N}] * b(k,l) \right) \right) \cdot \\ &\cdot \frac{\partial}{\partial y(p,q)} \left(\sum_r \sum_s y(r,s) b(k - Mr, l - Ns) \right) \Bigg) = \\ &-2 \sum_k \sum_l \left(f(k,l) - [y(k,l)]_{\uparrow M \uparrow N} * b(k,l) \right) \cdot \\ &\cdot b(k - Mp, l - Nq) = 0 \end{aligned} .$$

Defining the reversed (mirrored) reconstruction function with $b_r(i,j) = b(-i. -j)$, it is evident that previous expression may be written as:

$$\begin{aligned} & f(Mp,Nq) * b_r(Mp,Nq) = \\ & [y(Mp,Nq)]_{\uparrow M \uparrow N} * b(Mp,Nq) * b_r(Mp,Nq) \end{aligned}$$

Here, equation hold only at points (Mp, Nq). So, if we down-sample previous condition by factors M and N in x and y directions, we obtain:

$$[f(p,q) * b_r(p,q)]_{\downarrow M \downarrow N} = y(p,q) * [b(p,q) * b_r(p,q)]_{\downarrow M \downarrow N}$$

One of the conditions on $b(k,l)$ is that $[b(p,q) * b_r(p,q)]_{\downarrow M \downarrow N}$ has the inverse, so finally we have:

$$\begin{aligned} y(p,q) = & \left([b(p,q) * b_r(p,q)]_{\downarrow M \downarrow N} \right)^{-1} * \\ & * [f(p,q) * b_r(p,q)]_{\downarrow M \downarrow N} \end{aligned} \quad (1)$$

This is an expression which holds generally. These results suggest a simple procedure for the determination of the prefilters. Methodology is as follows. First prefilter input sequence $f(p,q)$ with chosen reversed reconstruction kernel $b_r(p,q)$,and down-sample the result. Second step is to prefilter obtained result with $\left([b(p,q) * b_r(p,q)]_{\downarrow M \downarrow N} \right)^{-1}$. After that step sequence is ready for reconstruction, resampling or calculation of derivative.

III. Reduction of the Computational Cost

In most practical cases, reconstruction filter $b(k,l)$ is given as a separable sequence: $b(k,l) = b(k)b(l)$. This fact could be exploited to reduce computational costs. To show how, few general expressions must be established first.

A. *Using the Property of Separation*

If we have separable function $f(i,j) = f_x(i) f_y(j)$, than convolving it with general sequence $g(i,j)$ gives:

$$f(i,j) * g(i,j) = f_x(i) * \left(f_y(j) * g(i,j) \right). \quad (2)$$

Furthermore, if we have $g(i,j) = g_x(i) g_y(j)$, then convolving it with separable $f(i,j)$ produces:

$$\begin{aligned} f(i,j) * g(i,j) &= \left(f_x(i) * g_x(i) \right) \left(f_y(j) * g_y(j) \right) \\ &= \varphi(i) \gamma(j), \end{aligned} \quad (3)$$

i.e. the result is again separable sequence. Next, we are interested in expression for the inverse sequence of separable sequence $\chi(i,j) = \varphi(i)\gamma(j)$.

B. *Inverse of the Separable Sequence*

Using (2) we may write:

$$\begin{aligned} \chi(i,j) * \chi^{-1}(i,j) &= \varphi(i) * \left(\gamma(j) * \chi^{-1}(i,j) \right) = \\ &= \delta(i,j) = \delta(i)\delta(j). \end{aligned}$$

Convolving previous expression first with $\varphi^{-1}(i)$ then with $\gamma^{-1}(j)$ leads to:

$$\gamma(j) * \chi^{-1}(i,j) = \varphi^{-1}(i)\delta(j) \;\Rightarrow\; \chi^{-1}(i,j) = \varphi^{-1}(i)\gamma^{-1}(j),$$

So we may conclude:

$$\chi^{-1}(i,j) = \left(\varphi(i)\gamma(j) \right)^{-1} = \varphi^{-1}(i)\gamma^{-1}(j). \quad (4)$$

C. *Optimized Computation Efficient Expression*

One more property must be established for down-sampling of separable functions. It is trivial to see that we may write:

$$\begin{aligned} [f(i)g(j)]_{\downarrow M \downarrow N} &= \left[[f(i)g(j)]_{\downarrow M \downarrow 1} \right]_{\downarrow 1 \downarrow N} = \\ \left[[f(i)]_{\downarrow M}\, g(j) \right]_{\downarrow 1 \downarrow N} &= [f(i)]_{\downarrow M} \, [g(j)]_{\downarrow N} \end{aligned} . \quad (5)$$

Returning to our separable reconstruction kernel $b(i,j)$, we see that its reverse is also separable: $b_r(i,j) = b_r(i) b_r(j)$. Because of that, their convolution is also separable (using (3)):

$$b(i,j) * b_r(i,j) = \left(b(i) * b_r(i) \right) \left(b(j) * b_r(j) \right).$$

Down-sampling of this expression using (5), gives rise to:

$$[b(p,q) * b_r(p,q)]_{\downarrow M \downarrow N} = [b(p) * b_r(q)]_{\downarrow M} \, [b(p) * b_r(q)]_{\downarrow N} .$$

To move further from (1), inversion of the previous expression must be calculated. Using (4), we can directly write:

$$\left([b(p,q)*b_r(p,q)]_{\downarrow M\downarrow N}\right)^{-1}=$$
$$=\left([b(p)*b_r(q)]_{\downarrow M}\right)^{-1}\left([b(p)*b_r(q)]_{\downarrow N}\right)^{-1}.$$

Substituting in (1) and using (2) leads to:

$$y(p,q)=\left([b(p)*b_r(p)]_{\downarrow M}\right)^{-1}*$$
$$*\left(\left([b(q)*b_r(q)]_{\downarrow N}\right)^{-1}*[b_r(p)*(b_r(q)*f(p,q))]_{\downarrow M\downarrow N}\right).$$

Finally, using (5) we obtain expression we have been searching for:

$$y(p,q)=\left([b(p)*b_r(p)]_{\downarrow M}\right)^{-1}*$$
$$*\left[b_r(p)*\left(\left([b(q)*b_r(q)]_{\downarrow N}\right)^{-1}*[b_r(q)*f(q,p)]_{\downarrow 1\downarrow N}\right)\right]_{\downarrow M\downarrow 1}.$$

This expression directly shows two-dimensional implementation. Using the notation $\psi_M(p)=$ $=([b(p)*b_r(p)]_{\downarrow M})^{-1}$ and similary $\psi_N(q)$ for prefilters, previous expression becomes:

$$y(p,q)=\psi_M(p)*[b_r(p)*(\psi_N(q)*[b_r(q)*f(q,p)]_{\downarrow 1\downarrow N})]_{\downarrow M\downarrow 1}$$

After calculating prefilters $\psi_M(p)$ and $\psi_N(q)$, first step is down-sampling of $b_r(q)*f(p,q)$ in y direction by factor N. This is followed by separable convolution with $\psi_N(q)$ in y and $b_r(p)$ in x direction, which produces new two-dimensional sequence that must be down-sampled in x direction by factor M. Finally, convolution of this sequence with $\psi_M(p)$ only in x direction yields coefficients $y(p,q)$. This procedure is shown in Fig 1, with simplification $M=N$ for sake of clearness.

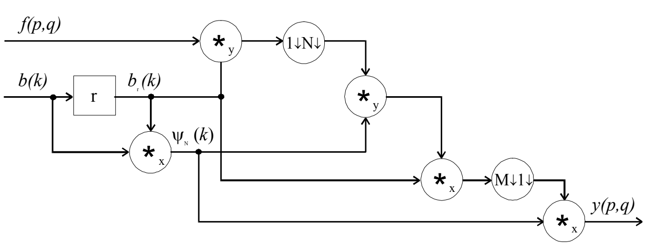

Fig. 1. Implementation of the least square design of the prefilters.

It must be stressed that such "separable" implementation is procedurally more complicated than direct (brute-force) implementation (1), but has significantly smaller complexity. Also, this "separable" scheme cannot be used if the reconstruction filter $b(i,j)$ is not separable.

IV. Results

Fig. 2 provides a comparison between three reconstruction kernels: Catmull Rom reconstruction kernel, cubic B-spline and Gauss function. On the left side of the Fig. 2 two-dimensional examples are shown and on the right side reconstruction is done on the three dimensional test function. The three-dimensional test function is proposed by Marschner and Lobb [3].

On the two-dimensional examples artefacts are mostly noticeable on the strips on the chest of the object. Reconstruction with Catmull Rom spline exhibit strong artefacts, while B-spline shows better result. Table 1 shows the variance between original and reconstructed object using three different reconstruction kernels. This table corresponds to the Figure 2 and supports the achieved results from the figure.

TABLE I
VARIANCES BETWEEN ORIGINAL AND RECONSTRUCTED OBJECT

Variance	2-dimension	3-dimension
Catmull-Rom	410	55.77
B-spline	389	53.12
Gauss	377	52.15

On the three-dimension examples when Catmull Rom spline is used artefacts are obvious on the crest of the waves. The B-spline shows smaller blinking on the crest. Reconstruction with the Gauss produces the best results. It must be accentuate that Gauss is wider then B-spline. If space limitation is applied on the reconstruction kernels B-spline is the best among the used filters.

V. Conclusion

The main objective of this paper has been to derive methodology for design of the prefilters required for any separable reconstruction function. This approach is interesting conceptually for the new insight in the prefilterig required for the minimal square error down-sampling and reconstruction procedure. We have shown that proposed design verify chosen reconstruction filters, so two-dimensional and three-dimensional examples confirm the numerical results.

References

[1] M. J. Bentum, B. A. Lichtenbelt, T. Malzbender, "Frequency Analysis of Gradient Estimators in Volume Rendering"', *IEEE Transactions on Visualization*, Vol. 2, No. 3, September 1996, pp. 242-253.

[2] R. Machiraju and R. Yagel, "Reconstruction Error Characterization and Control: A Sampling Theory Approach", *IEEE Transactions on Visualization and Computer Graphics*, Vol. 2, No. 4, December 1996, pp. 364-376.

[3] S. R. Marschner and R. J. Lobb, "An Evaluation of Reconstruction Filters for Volume Rendering", *Proc. Visualization '94., IEEE CS Press*, October 1994, pp. 100-107.

[4] D. P. Mitchell and A. N. Netravali, "Reconstruction Filters in Computer Graphics", *Computer Graphics*, Vol 22, No. 4, August 1988, pp. 221-228.

[5] T. Moller, R. Machiraju, K. Mueller and R. Yagel, "Evaluation and Design of Filters Using a Taylor Series Expansion", *IEEE Transactions on Visualization and Computer Graphics*, Vol. 3, No. 2, June 1997, pp. 184-199.

[6] M. Unser, A. Aldroubi and M. Eden, "Fast B-spline Transforms for Continuous Image Representation and Interpolation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 13, No. 3, March 1991, pp. 821-833.

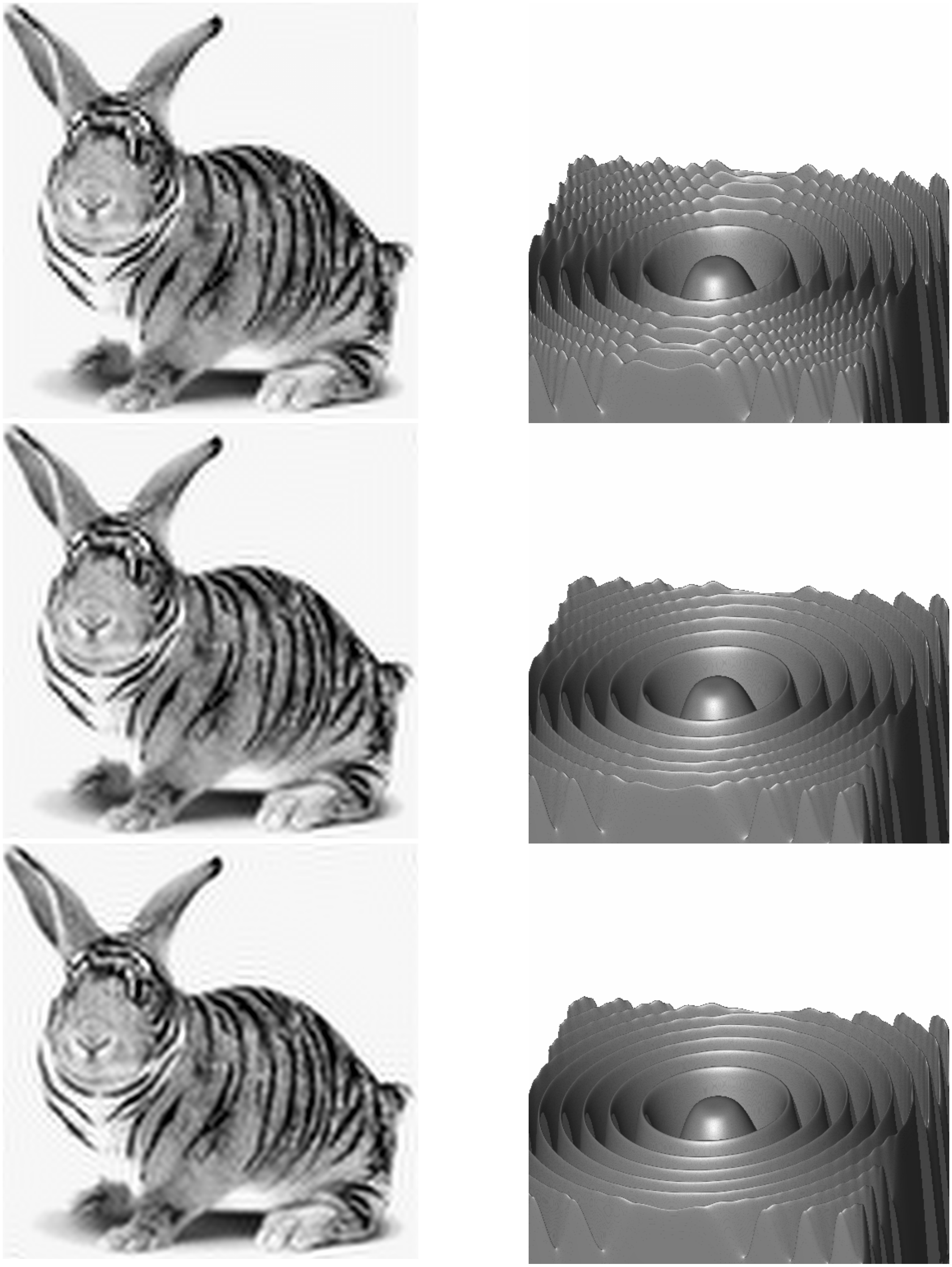

Fig. 1: Two-dimensional (left) and three-dimensional (right) examples of reconstruction. Three reconstruction kernels are used: Catmull-Rom spline, cubic B-spline, Gaussian. The original image 512^2 is down-sampled six times, and than reconstructed to initial size. Size of the volume is 128^3. It is down-sampled two times in each direction and reconstructed using same three reconstruction kernels as two-dimensional examples.